

(19)



(11)

EP 4 207 003 A1

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:

05.07.2023 Bulletin 2023/27

(21) Application number: **22210980.3**

(22) Date of filing: **02.12.2022**

(51) International Patent Classification (IPC):

G06N 5/022 (2023.01) **G06N 20/00** (2019.01)
G06N 5/04 (2023.01) **G05B 19/042** (2006.01)
G06N 3/0442 (2023.01) **G06N 5/01** (2023.01)
G06N 20/20 (2019.01) **G06N 3/09** (2023.01)

(52) Cooperative Patent Classification (CPC):

G06N 5/022; G05B 13/027; G05B 19/042;
G05B 23/024; G06N 3/0442; G06N 3/09;
G06N 5/01; G06N 20/20

(84) Designated Contracting States:

AL AT BE BG CH CY CZ DE DK EE ES FI FR GB
GR HR HU IE IS IT LI LT LU LV MC ME MK MT NL
NO PL PT RO RS SE SI SK SM TR

Designated Extension States:

BA

Designated Validation States:

KH MA MD TN

(30) Priority: **31.12.2021 US 202163295625 P**
28.04.2022 US 202217731659

(71) Applicant: **Schneider Electric Systems USA, Inc.**
Foxborough, MA 02035 (US)

(72) Inventors:

- **Gondhi, Dinesh**
500085 Hydernagar Medchal (IN)

• **Sinha, Bhaskar**

500019 Hyderabad (IN)

• **Patil, Ashish Bhaskarrao**

411045 Pune (IN)

• **Mahendran, Niranjana**

600040 Chennai (IN)

• **Namburu, Kalyana Srinivas**

500075 Hyderabad (IN)

• **Gavande, Vinod Krushna**

421201 Dombivli-East (IN)

(74) Representative: **Manitz Finsterwald**

Patent- und Rechtsanwaltspartnerschaft mbB

Martin-Greif-Strasse 1

80336 München (DE)

(54) **SYSTEMS AND METHODS FOR BUILDING A KNOWLEDGE BASE FOR INDUSTRIAL CONTROL AND DESIGN APPLICATIONS**

(57) A method of automating engineering design is provided. The method includes receiving a training set including pairings of control loop data for respective control loops identified in digitized design data and templates that were instantiated using the control loop data of the respective control loops and training, using machine learning, a knowledge base, based on the training set.

The knowledge base, once trained, is configured to be queried with digitized new control loop data, predict a template to pair with the digitized new control loop data, and the predicted template, and the predicted template is configured to be instantiated with the new control loop data for implementation of a control loop in an engineering system.

EP 4 207 003 A1

Description

TECHNICAL FIELD

[0001] The present disclosure relates to engineering automation, and more particularly, to knowledge driven artificial intelligence engine for engineering automation for industrial control and design applications.

BACKGROUND

[0002] The design of specifications for control applications, human machine interfaces (HMIs), and cabinet engineering for a particular process are performed manually. The effort utilizes significant time and resources to understand characteristics of the process from the overall design inputs and requirements. The overall design inputs can include, for example, a front-end engineering and design (FEED) that can include, e.g., piping and instrumentation diagrams (P&IDs) and a database of component identifiers (also referred to as TAGs)). When developing a new design, engineers rely on skill and can use significant amounts of time to determine if a new solution should be created or if a specific requirement has already been addressed by a previous team that can be leveraged by reusing associated existing artifacts.

[0003] When this problem is approached by applying a strictly rules driven approach, there are constraints that limit use and a lack of scalability. The process still requires a high level of engineering skill and significant time resources to design the rules for every new project and each type of project.

[0004] While conventional methods and systems for monitoring dangerous conditions in an operation unit have generally been considered satisfactory for their intended purpose, there remains a need for improvements.

SUMMARY

[0005] The purpose and advantages of the below described illustrated embodiments will be set forth in and apparent from the description that follows. Additional advantages of the illustrated embodiments will be realized and attained by the devices, systems and methods particularly pointed out in the written description and claims hereof, as well as from the appended drawings. To achieve these and other advantages and in accordance with the purpose of the illustrated embodiments, in one aspect, disclosed is a method of automating engineering design. The method includes receiving a training set including pairings of control loop data for respective control loops identified in digitized design data and templates that were instantiated using the control loop data of the respective control loops and training, using machine learning, a knowledge base, based on the training set. The knowledge base, once trained, is configured to be queried with digitized new control loop data, predict a template to pair with the digitized new control loop data,

and the predicted template, and the predicted template is configured to be instantiated with the new control loop data for implementation of a control loop in an engineering system.

[0006] In one or more embodiments, the knowledge base includes a machine learning model, and training the machine learning model can include training a classifier of the machine learning model.

[0007] In one or more embodiments, the received digitized control loop data and the received templates can be standardized and/or normalized.

[0008] In one or more embodiments, the method can further include submitting the pairings for a conflict review performed manually and/or automatically, receiving review data based on the conflict review, and updating the control loop data in the pairings as a function of the conflict review data, wherein the knowledge base is trained using the updated pairings.

[0009] In one or more embodiments, the conflict review can include identifying a conflict in which first and second control loop data for two different pairings are the same and are paired respectively with different templates, reviewing the digitized design data to identify an additional attribute of the respective first and second control loops that is different for the first control loop relative to the second control loop, and including in the review data a new feature to be added to the control loop data for the first and second control loops that corresponds to the additional attribute so that the first and second control loops have different corresponding control loop data.

[0010] In one or more embodiments, the method can further include adding the additional attribute to the control loop data of at least one of the first and second control loops.

[0011] In one or more embodiments, the method can further include receiving feedback about pairings between digitized new control loop data and predicted templates output by the knowledge base in response to queries submitted to the knowledge base and updating the training set based on the feedback.

[0012] In one or more embodiments, the knowledge based can further be configured to adjust a confidence score associated with the predicted template based on the feedback.

[0013] In accordance with further aspect of the disclosure, an automated engineering system is provided that includes a memory configured to store instructions and a processor in communication with the memory, wherein the processor upon execution of the instructions is configured to perform the disclosed method.

[0014] In accordance with still further aspects of the disclosure a non-transitory computer readable storage medium and one or more computer programs embedded therein is provided, which when executed by a computer system, cause the computer system to perform the disclosed method.

[0015] These and other features of the systems and methods of the subject disclosure will become more read-

ily apparent to those skilled in the art from the following detailed description of the preferred embodiments taken in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] A more detailed description of the disclosure, briefly summarized above, may be had by reference to various embodiments, some of which are illustrated in the appended drawings. While the appended drawings illustrate select embodiments of this disclosure, these drawings are not to be considered limiting of its scope, for the disclosure may admit to other equally effective embodiments.

FIG. 1 is a block, flow diagram illustrating an example artificial intelligence (AI)-based design system for AI based automated engineering project design, in accordance with embodiments of the invention;

FIG. 2, is a schematic diagram illustrating an example template for a control loop, in accordance with embodiments of the invention;

FIG. 3, is a schematic diagram illustrating an example template in an uninstantiated state and in an instantiated state, in accordance with embodiments of the invention;

FIG. 4 is a block, flow diagram illustrating an example knowledge base builder/trainer engine of the AI-based design system, in accordance with embodiments of the invention;

FIG. 5, illustrates example entries for pairings stored in a collection of pairings of a knowledge base builder/trainer engine of the AI-based design system, in accordance with embodiments of the invention;

FIG. 6, illustrates example entries for pairings stored in a collection of pairings of a knowledge base builder/trainer engine of the AI-based design system when a conflict is detected before and after resolution is performed, in accordance with embodiments of the invention, in accordance with embodiments of the invention;

FIG. 7, is a block, flow diagram illustrating example machine learning aspects of training knowledge base builder/trainer engine, in accordance with embodiments of the invention;

FIG. 8, is a block, flow diagram illustrating application of a knowledge base in the AI-based design system for making predictions, in accordance with embodiments of the invention;

FIG. 9, is a flowchart illustrating an example method

of querying a knowledge base of the AI-based design system, in accordance with embodiments of the invention;

FIG. 10 is a flowchart illustrating an example method of training a knowledge base of the AI-based design system, in accordance with embodiments of the invention; and

FIG. 11 is a block diagram of an example computer system used for implementation of the AI-based design system, in accordance with embodiments of the invention.

[0017] Identical reference numerals have been used, where possible, to designate identical elements that are common to the figures. However, elements disclosed in one embodiment may be beneficially utilized on other embodiments without specific recitation.

DETAILED DESCRIPTION

[0018] With reference now to the drawings, for purposes of explanation and illustration, and not limitation, a block diagram of an exemplary embodiment of a design system in accordance with the disclosure is shown in FIG. 1, wherein the design system is designated generally by reference character 100. Methods associated with monitoring one or more operation units with design system 100 in accordance with the disclosure, or aspects thereof, are provided in Figs. 2-11, as will be described.

[0019] It is noted that FIGS. 1, 4, and 7-10 show flow diagrams and flowcharts. The flows in FIGS. 1, 4, and 7-10 demonstrate implementation of various exemplary embodiments. It is noted that the order of operations shown in FIGS. 1, 4, and 7-10 is not required, so in principle, the various operations may be performed out of the illustrated order. Also certain operations may be skipped, different operations may be added or substituted, some operations may be performed in parallel instead of strictly sequentially, or selected operations or groups of operations may be performed in a separate application following the embodiments described herein.

[0020] Language that refers to the transfer of information is not meant to be limiting. For example, the term "receive" as used herein refers to obtaining, getting, accessing, retrieving, reading, or getting a transmission. Use of any of these terms is not meant to exclude the other terms. Data that is transferred to or from a module can be transferred by a transmission to or from the module, or can include the data in a location that can be accessed by the module or is provided in a manner to be accessible to another module.

[0021] Design system 100 is an artificial intelligence (AI)-based design system configured to automate design of engineering workflows, such as control applications, human-machine interface (HMI) design and cabinet design. Design system 100 includes a knowledge base (KB)

builder/trainer engine 102, a KB 104, a mapping engine 106, a configuration engine 108, a new design data manager 110, a user review interface 120, and an engineer review interface 122. Design system 100 receives input data from a past engineered projects database 130 which is used to build KB 104.

[0022] Design system 100 further receives review data via a review interface 120 and/or engineer review interface 122. The review data can be requested and/or generated automatically or manually. The review data is used to update the KB 104, such as for updating mappings of templates, updating control loop data, updating confidence scores associated with mappings between templates and control loops, and/or adding new templates that can be used by KB 104 for mappings.

[0023] A control loop performs some sub-process in the plant process and may be either a closed or an open control loop. Most control loops involve several devices working together, including sensors, actuators, controllers, and the like. There are generally two types of control loops: simple control loops, and complex control loops. Simple control loops have an input, an output, and a controller, while complex control loops are made of several simple control loops.

[0024] Design system 100 further receives new project data 132 with a request to automatically generate configuration data for an engineering workflow. The configuration data is output to control configuration database 134 where it can be applied for designing an engineering system 170.

[0025] With reference to architecture of design system 100 and its related storage, each of its modules, including KB builder/trainer engine 102, KB 104, mapping engine 106, configuration engine 108, new design data manager 110, user review interface 120, and engineer review interface 122, is implemented using a physical or virtual computing device having a central processing unit (CPU), random access memory (RAM), and a storage medium, which can be connected through buses. Programmable instructions can be stored in the storage medium and executed by the CPU to cause the CPU to perform operations described herein. Whether implemented as physical or virtual device(s), the modules of design system 100 use one or more local or remote hardware processing devices that execute software instructions, which enables performance of the disclosed functions.

[0026] One or more of KB builder/trainer engine 102, KB 104, mapping engine 106, configuration engine 108, new design data manager 110, user review interface 120, and engineer review interface 122 can be integrated with each other, can be implemented by discrete devices, or can share one or more hardware or software components. Functionality can be shared or divided amongst mapping engine 106, configuration engine 108, new design data manager 110, user review interface 120, and engineer review interface 122 in a variety of ways.

[0027] Communication between KB builder/trainer en-

gine 102, KB 104, mapping engine 106, configuration engine 108, new design data manager 110, user review interface 120, and engineer review interface 122 can use wired or wireless communication, including one or more buses, wireless links, and/or computer networks (such as an intranet, a local area network (LAN), and/or a wide area network (WAN)).

[0028] In the example shown, one instance is shown for each of KB builder/trainer engine 102, KB 104, mapping engine 106, configuration engine 108, new design data manager 110, user review interface 120, and engineer review interface 122, however one skilled in the art will recognize that the number of instances of these respective modules is not limited to a particular number.

[0029] KB 104 and storage components within respective modules KB builder/trainer engine 102, mapping engine 106, configuration engine 108, new design data manager 110, user review interface 120, and engineer review interface 122 can store data structures in volatile or non-volatile memory or on persistent storage (such as a file system). The storage component(s) of each of the modules KB builder/trainer engine 102, KB 104, mapping engine 106, configuration engine 108, new design data manager 110, user review interface 120, and engineer review interface 122 can be integrated with or accessible to the corresponding module and other modules of design system 100.

[0030] The functions of the modules KB builder/trainer engine 102, KB 104, mapping engine 106, configuration engine 108, new design data manager 110, user review interface 120, and engineer review interface 122 of design system 100 are now described together with the flow of design system 100.

[0031] The term knowledge base can refer to one or more ways of collecting, organizing, retrieving, and sharing knowledge, which can include, for example, application of rules, data mining, modelling, artificial intelligence, machine learning, or the equivalent.

[0032] Starting with flows 151-154, building or training of KB 104 is described. Past engineered projects (PEP) database 130 stores and submits digitized design data that was obtained from past engineered projects. The past engineered projects can be provided as documents in formats such as pdf, png, xlsx etc., including pictorial and/or textual data. Pictorial and textual data can include, for example, diagrams, charts, alphanumeric data, photographs, etc. The documents can include, for example, piping and instrumentation diagrams (P&IDs), process flow diagrams (PFDs), cause and effect (C&E), a tag database etc. The digitized design data obtained from the design data of past engineered projects is referred to as past digitized design data. The past engineered projects include instantiated template(s), wherein the digitized design data for the past engineered projects was used to instantiate one or more templates (referred to as past templates). Identification of the past digitized design data and the past templates with which it was instantiated are paired together and used to build or train KB 104.

[0033] With additional reference to FIG. 2, an example template 200 is shown. A template is a software program or logic used by a distributed control system (DCS) system to perform specific functions, such as process control, data acquisition, and alarming. This program is executed by a controller (a DCS hardware component) that can be disposed, for example, in an industrial plant. This program can be represented by a group of interconnected function blocks 202, such as one or more input blocks 202A, one or more control and/or calculation blocks 202C, and one or more output blocks 202B. The input block(s) 202A are configured to acquire data (e.g., temperature, pressure, flow, level, etc. from a field device), and the output block(s) 202B are configured to output a control signal to control a physical entity (e.g., to control steam inflow to a vessel by manipulating a valve position, to control electrical flow by manipulating a switch, etc.). The control block(s) 202C can include, for example, a proportional-integral-derivative (PID) control block, logic, mathematical operations, etc. The control and/or calculation block(s) 202C are configured to determine the control signal for manipulating a physical entity in order to control a process variable (e.g., temperature, pressure, flow, level, etc.) to a desired value or within a desired range by.

[0034] With returned reference to FIG. 1, an instantiated template (whether past template or new template) is an instantiation of the template in which selected digitized design data was inserted in fields of the template. The instantiation of past templates with past digitized design data can be performed manually, as automatic instantiation of templates with digitized design data would be unavailable for past engineering projects without the systems provided by the disclosed features of KB builder/trainer engine 102, new design data manager 110, and mapping engine 106. Digitization of past engineering projects for providing the past digitized design data is disclosed by U.S. Patent Nos. 10671894 and 10635945.

[0035] With additional reference to FIG. 3, shown are an example sample 304 of digitized data, an example instance 302A of a template in an uninstantiated state, and an example instance 302B of the template in an instantiated state. The sample 304 shows digitized data entries for four control loops, with an entry corresponding to each tag included in each of the loops. The digitized entries include control loop data, including control loop names to identify the respective control loops, tag names to identify the corresponding tag, system type (e.g., field-bus foundation or I/O), input/out (I/O) type, signal level, and/or signal description. Additional, less, or different control loop data can be included with the stored digitized data. All entries of digitized data that have the name ControlLoop1 are used for instantiating the template, resulting in the instantiated instance 302B. Namely, the entry for tag name LT1001 was used to instantiate an input block and the entry for tag name CV1001 was used to instantiate an output block.

[0036] With returned reference to FIG. 1, at flow 151,

past templates instantiated with past digitized design data for a past project, including a pair of identification of the past template (template ID) and identification of the past digitized design data (past digitized design data ID), are provided to KB builder/trainer engine 102. KB builder/trainer engine 102 identifies past control loops in the past digitized design data (referred to as past control loops). Identification of control loops is described in U.S. patent applications having serial numbers 17/442,403, 17/442,423, and 17/442,437, the contents of which are incorporated herein in their entirety. Each past control loop has associated control loop data that describes attributes of the control loop, such as control loop type, one or more tags, and attributes of the tags (e.g., such as system type, signal type, signal count, signal level, alarms, and/or signal and/or equipment description, etc.).

[0037] A tag of a control loop is a unique identification name given to a device or an equipment. A control loop extracted from past or new digitized design data includes a logical group of tags performing a specific control operation. For example, an control loop for measuring temperature of a fluid in a vessel and doing a control operation, e.g., controlling a valve to supply the steam as needed to the vessel in order to maintain the temperature of the fluid in the vessel at a certain setpoint can include a first tag named Tag1 for measuring the temperature and a second tag named Tag2 for controlling the operation.

[0038] Once a past control loop is identified, a determination can be made regarding which past template the past control loop was used for instantiation. A past control loop can be used to instantiate one or more past templates or multiple control loops can be used to instantiate one past template. Pairings are determined that include the identification (past control loop data ID) and past control loop data associated with each past control loop and identification of each past template (template ID) involved in an instantiation.

[0039] With reference to FIGS. 1 and 5, a number of entries is shown as sample 500 for pairings stored in collection 404 by KB builder/trainer engine 102. Identifiers of past control loops are shown in column 502, corresponding features of control loop data is shown in columns 504, corresponding past template identification is shown in column 506, confirmation regarding whether the entry is conflict free is shown in column 508, and a tool to add a feature to the control loop data is shown in column 510. In the example shown, in a first entry, control loops identified as ControlLoop1 and ControlLoop2 have the same control loop data in corresponding features (as shown in columns 504), are paired with the same template (which is identified in column 506), are confirmed as being conflict free, and do not have any added features. The same is true for the control loop(s) identified in the remaining entries shown.

[0040] With additional reference to FIG. 4, in one or more embodiments, at flow 152 KB builder/trainer engine 102 submits the pairings stored in entries of collection

404 to review interface 120 for detection and/or resolution of conflicts in the pairings. In one or more embodiments, training engine 406 detects conflicts in entries of collection 404 and submits the conflicts to review interface 120 for manual or automated resolution. The division of functions performed by review interface 120 and training engine 406 is not limited to a specific design, and detection and resolution can be performed by either of these modules. Detection and resolution of conflicts is explained with reference to FIG. 6 and the related description.

[0041] In one or more embodiments, review interface 120 can provide an interface to an expert user, such as a subject matter expert (SME). In one or more embodiments, review interface 120 can provide an interface to an automated process for detecting and/or resolving conflicts. Review data including a resolution of a detected conflict is provided (by the SME or automated process) via review interface 120 at flow 153. KB builder/trainer engine 102 receives the resolution and updates collection 404.

[0042] With reference to FIGS. 1 and 6, a number of entries is shown as sample 600A for pairings stored in collection 404 by KB builder/trainer engine 102 when a conflict is detected before resolution. The same entries are shown as sample 600B after the detected conflicts have been resolved. In sample 600A, the respective entries for ControlLoop8 and ControlLoop9 have the same control loop data, but are paired with different templates, which established a conflict that has been flagged in column 508. The conflicting entries can optionally be highlighted or otherwise displayed with a graphical indicator of the conflict. During the resolution process the corresponding past digitized design data (or original past design data prior to digitization) was consulted for additional details about the control loops that would differentiate ControlLoop8 and ControlLoop9 that would account for assignment of a different template to these respective control loops.

[0043] The consultation (manual or automatic) revealed that ControlLoop8 and ControlLoop9 have different signal identifiers. In sample 600B, it is shown that via activation of the add feature tool at column 510, manual or automatic addition of column 512 was allowed to the columns of control loop data 502. The confirmation that the entries for ControlLoop8 and ControlLoop9 are conflict free was updated.

[0044] With returned reference to FIG. 1, pairings of identification of past control loops (control loop ID) and the associated past control loop data, with past templates used in the same instantiation are provided to KB 104 at flow 154. Flow 154 can further include adjustments to the pairings, namely based on adjustment to the control loop data used to identify a past control loop in accordance with resolutions received at flow 153 from review interface 120. While flow 154 is described as including pairings and adjustments to pairings, flow 154 can include any data used to build or train KB 104, such as a classifier trainer 702, as described in greater detail with

reference to FIG. 7.

[0045] KB builder/trainer engine 102 standardizes and/or normalizes the past control loop data and past templates before providing them to KB 104 (as described with respect to identification engine 402 of KB builder/trainer engine 102 in FIG. 4).

[0046] KB 104 is revised by storing mappings for each of the pairings received via flow 154 as well as resolved, new, or updated pairings received via flow 165. New and updated pairs are described in greater detail with respect to the description of flows 163A and 163B. The mappings include the identification and control loop data of the past control loop(s) in the pairing as mapped to the past template identified in the pairing. In addition, each mapping can have an associated confidence score. Upon an initial storing of a pairing in KB 104, e.g., as a mapping, the confidence score can be neutral. Each time a pairing for the same mapping is submitted again to KB 104 at flow 154, the confidence score is increased.

[0047] While the pairings received at flow 154 and mappings stored by KB 104 are described as 1:1 mappings with control loop data for one control loop correlated to one template in the pairings and mappings, one skilled in the art will recognize that control loop data for respective multiple control loops can correspond to one template, control loop data for one control loop can correspond to multiple templates, or control loop data for multiple control loops can correspond to multiple templates. Although the other flows may refer to control loop data for one control loop or template ID of one template, it is understood that either of these could be substituted with multiple instances.

[0048] Each time a conflict is detected, confidence scores can be updated to reflect the conflict and the resolution. For example, if a mapping with the conflict exists in KB 104, its associated confidence score can be decreased or the mapping can be removed. If a mapping with the revised control loop data (in accordance with the resolution) exists in KB 104, its associated confidence score can be increased.

[0049] With reference to FIGS. 1 and 4, KB builder/trainer engine 102 can include an identification engine 402, a collection 404, and a training engine 406. Identification engine 402 receives the past templates with past digitized design data that was used for instantiating the past templates, extracts the past control loops from the past digitized design data, and determines the pairings. The pairings of identifications of past control loops and associated past control loop data with past templates are stored in collection 404 at flow 410. Identification engine 402 can normalize and/or standardize the past control loop data and/or past templates before storing the pairings in collection 404. A transformation process can be applied for converting nonstandard values in the past control loop data into standard or normalized values. For example, a signal type of a tag attribute can be included in the past control loop data as "AI." Identification engine 402 can standardize or normalize the signal type to be

"Analogue Input". A similar process is used for templates. Two templates are similar can be automatically recognized, with one of the two templates used as a member of the mapping pair.

[0050] Collection 404 is updated based on resolutions by review interface 120. Training engine 406 trains (or builds) KB 104 by providing the pairings (or trained data) from collection 404 at flow 154 with any revisions performed to KB 104 and updating KB 104 by storing a mapping for each pairing provided at flow 154.

[0051] With returned reference to FIG. 1, KB 104 uses the pairings to provide an association between control loop data associated with control loops and templates to which the respective control loops were paired and mapped. The control loops can be identified from past digitized design data or new digitized data (based on feedback).

[0052] New design data 132 that corresponds to a new engineering product can be submitted at flow 155 to new design data manager 110. New design data 132 can be implemented in an engineering system, such as engineering system 170. Flows 155-161 and flow 162A illustrate application of KB 104 for receiving the new design data and using a prediction about a template for mapping of the new design data in order to generate control configuration data that can be used to design engineering system 170.

[0053] New design data 132 is received via flow 155 by design data manager 110 in pictorial and/or textual format. Design data manager 110 digitizes the new design data. Digitization of design data is described is disclosed by U.S. Patent Nos. 10671894 and 10635945. Design data manager 110 further identifies control loops and corresponding control loop data in the new design data (also referred to as new control loops and new control loop data). Design data manager 110 can further normalize and/or standardize the new control loop data. The method used by design data manager 110 for identifying control loops can be the same or similar to the process used by KB builder/trainer engine 102 for identifying control loops. The identified new control loops are provided via flow 156 to mapping engine 106. The identified new control loop data includes, for example, control loop type and tags belonging to the control loop along with attributes of the tags.

[0054] Design data manager 110 provides the new control loops to mapping engine 106. Mapping engine 106 queries KB 104 at flow 157 for templates to use for each new control loop. The query for a new control loop includes the new control loop data for the new control loop. KB 104 receives the query and selects the template that has associated control loop data that matches the control loop data in the query. The matching process is simplified by the normalization and/or standardization of the control loop data in the query and the control loop data stored in KB 104. If KB 104 selects more than one template for the new control loop a first section of tags present in the new control loop data can be mapped to

one of the templates and a second section of tags of the new control loop data can be mapped to a different one of the templates. A different section of tags can be mapped to each of the respective different templates.

The selected template is provided to mapping engine 106 at flow 158 as a recommendation for pairing with the control loop data. The control loop data of the new control loop can thus be mapped to the selected template.

[0055] Once a template is selected for a new control loop, mapping engine 106 maps the selected template with new control loop data of the new control loop and provides the mapping (which identifies the selected template and the new control loop) to configuration engine 108 at flow 161. In one or more embodiments, mapping engine 106 submits at flow 160 the mapping to an engineer review interface 122. In one or more embodiments, engineer review interface 122 can provide an interface to an expert user, such as a project engineer. The project engineer reviews the mapping for appropriateness, e.g., from a practicality, practicability, and/or engineering point of view. The project engineer may provide review data as feedback at flow 160B via engineer review interface 122 that the mapping is disapproved (e.g., for being incorrect or not ideal) or approved. In addition, the review data provided at flow 160B by the project engineer may include a suggestion with a replacement mapping that identifies a replacement template to use instead of the selected template.

[0056] If the mapping was disapproved and a replacement template was not suggested, the method continues at flow 161. If the mapping was disapproved and a replacement template was suggested, the identified replacement template is retrieved from template library 149, and the method continues at flow 161. If the replacement template as identified in the review data is not found in template library 149, the method continues at flow 161. If the mapping was approved, the selected template is retrieved from template library 149, and the method continues at flow 161.

[0057] At flow 161 the instantiated selected or replacement template are provided to configuration engine 108. Furthermore, in one or more embodiments, the review data and success status of retrieving the selected or replacement template is reported to configuration engine 108. The review data reported can include an indication of a disapproval by the project engineer and whether a replacement template was suggested.

[0058] Logic and functions performed by configuration engine 108 are shown at blocks 144 and 146. At block 144, a determination is made whether the mapping suggested by KB 104 at flow 158 was approved. If the mapping was approved, notification of the approval is provided to KB 104 at flow 162A1 with a command to increase the confidence score associated with the mapping. In addition, the mapping and the selected template (if successfully retrieved) are provided to configuration generator 142 at flow 162A2. Configuration generator 142 can instantiate the selected template using the new control

loop data.

[0059] If the mapping was disapproved, notification of the disapproval is provided to KB 104 at flow 162B1 with a command to decrease the confidence score associated with the mapping and the method continues at block 146, as indicated by flow 162B2.

[0060] At block 146, a determination is made whether the selected template or replacement template were available in template library 149. If the selected or replacement template was available in template library 149, at flow 163A2 the selected or replacement template is provided to configuration generator 142 with the new control loop (if not already provided at flow 162A2). Configuration generator 142 can instantiate the selected or replacement template using the new control loop data.

[0061] In addition, at flow 163A1, a request is provided to review interface 120 suggesting a modification for mapping the replacement template to the control loop (instead of the selected template), per the suggestion from engineering review interface 122 at flow 160B. If the selected or replacement template is not available in template library 149, at flow 163B, a request is provided to review interface 120 with a suggestion to add a new template to template library 149 as specified via engineering review interface 122 at flow 160B. While identification of the selected or replacement template to be added to template library 149 can be provided with human intervention, communication to review interface 120 can be automated.

[0062] At flow 164, review interface 120 provides template IDs of the modified or newly added template and control loop IDs of the control loops having control loop data with which it is to be mapped. At flow 165, KB builder/trainer engine 102 provides instructions to KB 104 to update any mapping it has that includes the control loop ID of the new control loop and template ID of the new or modified template.

[0063] Configuration generator 142 can handle updating control configuration database 134 with the mappings to the selected or replacement templates, and can further perform instantiation of the selected or replacement templates using the new control loops in accordance with the mapping. The instantiated templates can be stored in control configuration database 134. A control component of engineering system 170 can be operated using the instantiated templates in control configuration database 134 or can access the mappings in control configuration database and perform the instantiation, which is then utilized for operation of the control component.

[0064] As shown, a process of creating, building, and/or training KB 104 includes the process performed at flows 151, 152, 153, and 154. A process for making predictions using KB 104 includes flows 155, 156, 157, 158, 161, and 162A2. A self-evolving process for enhancing KB 104 includes flows 160A, 160B, 162A1, 162B1, 162B2, 163A1, 163A2, 163B, 164, and 165.

[0065] With reference to FIG. 7, shown are KB builder/trainer engine 102A, KB 104A, identification engine

402A, collection 404A, and ML model training engine 406A are shown while performing an example process of building or creating KB 104. The description of KB builder/trainer engine 102, KB 104, identification engine 402, collection 404, and training engine 406, including interactions between these components, as described with respect to FIGS. 1 and 4 is applied to builder/trainer engine 102A, KB 104A, identification engine 402A, collection 404A, and ML model training engine 406A, respectively. Thus, the description of model training engine 406 is relevant to ML model training engine 406A, but as its label indicates, ML model training engine 406A shows an embodiment that includes classifier trainer 702 and is specific to training a ML model 706 of KB 104A.

[0066] Identification engine 402A of KB builder/trainer engine 102A receives and normalizes inputs received via flows 151, 153, or 164. The inputs to KB 104A can be formatted with a specified shape. Example respective shapes can include batch size, steps, features) and activation function. In an example, the control loop data has a shape Nx15x21 (meaning batch size N, steps 15, features 21) and the template identified by template ID has a shape Nx32 (meaning batch size N, output features 32, and activation function). In an example, the activation function used is softmax (without limitation), which is a mathematical function for converting a vector of numbers into a vector of probabilities which can be applied to a neural network model.

[0067] In the example shown, the data provided via flow 151 is provided from past engineered projects database 130. The data included in the past engineered projects database 130 can include, for example, a name of a template paired with a control loop, and control loop data that is particular to the control loop, such as system type, input/output (I/O) types and/or names, signal levels, etc.

[0068] Once the inputs are normalized by identification engine 402A, collection 404A updates and aggregates its collection of information about pairings between control loop data and templates, forming a training set. Updates to collection 404A are communicated to ML model training engine 406A as a training set at flow 412A.

[0069] An example training set is shown in FIG. 7 that demonstrates how example actual data received via any of flows 151, 153, 164 is aggregated by tallying a number of values of attributes in different features of control data that is paired with an identified template.

[0070] ML model training engine 406A includes classifier trainer 702 for using the training set from collection 404A to build and train a classifier, such as random forest classifier, without limitation to a particular classifier. A random forest classifier is trained (e.g., by classifier trainer 702) to have multiple decision trees generated from random samples of data. Each tree provides certain predictions. A prediction output from the classifier is based on voting among the trees. The class with maximum votes is selected to be output as a prediction.

[0071] Classifier trainer 702 receives the training set

(which can include updates to the training set) via flow 412A. Classifier trainer 702 provides the trained classifier to ML model 706 via flow 708. ML model 706 can use the trained classifier to predict pairings between control data and templates.

[0072] In one or more alternative embodiments, KB builder/trainer engine 102 and KB 104A can build and/or use neural networks as an alternative to using classifier trainer 702. The neural networks can be built, for example, using network of long short-term memories (LSTMs).

[0073] ML model 706 can be configured to assign a confidence score to the predictions (predicted pairings) that it outputs. Upon an initial training of ML model 706 by the classifier trainer 702, the confidence score associated with predictions can be neutral. As a training set is updated with further instances of pairings between particular control loop data and a particular template, the confidence score for mapping the particular control loop data to the particular template for future predictions can be increased.

[0074] While the trained classifier provided at flow 708 can be trained to predict 1:1 mappings of control loop data for one control loop correlated to one template in the pairings and mappings, control loop data for respective multiple control loops to one template, control loop data for one control loop to multiple templates, or control loop data for multiple control loops to multiple templates. Although the other flows described may refer to control loop data for one control loop or template ID of one template, it is understood that either of these could be substituted with multiple instances.

[0075] Each time a conflict is detected, the classifier trainer 702 can update the trained classifier to adjust the confidence scores to reflect the conflict and the resolution. For example, upon detection of a conflict associated with a particular pairing of control data to template, the classifier trainer 702 can train the classifier to discontinue mapping the control data to the template or to decrease a confidence score associated with the mapping. Similarly, a mapping for revised control loop data (in accordance with the resolution) can be used to train the trained classifier to increase an associated confidence score.

[0076] ML model 706 can continue to be trained by classifier trainer 702 as new design data is received at flow 155 (shown in FIG. 1). Feedback about the prediction result output by ML model 706, such as feedback received via the engineer review process described in connection with FIG. 1 and flows 160A, 160B, 162A1, 162B1, 162B2, 163A1, 163B, and 164, can be used to update the training set provided at flow 412A and the trained classifier provided at flow 708. The confidence score can be adjusted based on feedback, including the feedback received by the engineering review process.

[0077] With reference to FIG. 8, KB 104A is shown, once ML model 706 is trained, while performing the process of making a prediction. In one or more embodiments, ML model 706 can apply neural networks, probabilistic algorithms, or other methods for making predictions of

which template to pair with the control data. An ML mapping engine 106A having a classifier predictor 802 is shown. The description of mapping engine 106, as described with respect to FIGS. 1 and 4 is applied to ML mapping engine 106A, but as its label indicates, ML mapping engine 106A shows an embodiment that includes classifier predictor 802 and is specific to handling and responding to prediction queries using machine learning.

[0078] ML mapping engine 106A receives identified new control loops via flow 156. The identified control loops can be input to classifier predictor as inputs 851. An example input is shown for an identified new control loop. At flow 157A, classifier predictor 802 of ML mapping engine 106A submits a prediction query to ML model 706 based on inputs 851. The prediction query includes multidimensional control loop data for one or more identified new control loops. The dimensional information in the query can be described, for example, by shape.

[0079] ML model 706 applies machine learning to predict which template would be paired to the control loop data in the query. For example, when ML model 706 is trained as a random forest classifier, the template predicted is selected based on voting among trees of the random forest classifier. The predicted template is output as a prediction result at flow 158A. The prediction result can also be described by shape.

[0080] Classifier predictor outputs identification of the predicted template as outputs 852. These outputs can be provided to configuration engine 108 via flow 161 and/or to engineer review interface 122 via flow 160A, as shown in FIG. 1.

[0081] With reference to FIG. 9, an example method of querying a KB is disclosed. At block 902, a KB (e.g., KB 104 in FIG. 1) is queried for a template to map with new control data that was identified in new digitized design data for a new engineering project. The query includes the new control loop data. The KB is trained to map past control loop data of past control loops that were identified in past digitized design data from one or more past engineering projects to respective templates based on past instantiation of the respective templates with the past control loops by the one or more past engineering projects. The new control loop data and the past control loop data can be standardized and normalized.

[0082] At block 904, a selected template is received in response to the query. The selected template is selected based on its mapping with past control loop data that matches the new control loop data. At block 906, configuration data is provided that includes an instantiation of the new control loop data with the selected template.

[0083] In one or more embodiments, at block 908, the KB is updated in accordance with feedback from a review process of the selected template with respect to the new control loop data in the query. For example, a mapping of the selected template to the new control loop data is submitted for an engineering review and the KB is updated as a function of review data received from a process of the engineering review. For example, a confidence

score associated with mappings in the KB that were approved by the engineering process can be increased. A confidence score associated with mappings in the KB that were disapproved by the engineering process and/or in which a modification to or replacement of the selected template was suggested by the engineering review, can be decreased. The mappings that were approved or disapproved are mappings having the same control data as the new control data and the same template as the selected template.

[0084] With reference to FIG. 10, an example method of training a KB is disclosed. At block 1002, a control loop ID and control loop data for respective control loops identified in digitized design data are received. At block 1004, template IDs of respective templates instantiated using corresponding control loops of the respective control loops are received.

[0085] At block 1006 pairings are formed of control loop IDs and control loop data with a template ID of the template that was instantiated using the corresponding control loop

[0086] At block 1008, the KB is trained, using machine learning, based on the pairings.

[0087] In one or more embodiments, at block 1010, the control loop data in the pairings is updated based on review data output by a conflict review. The conflict review can be performed manually and/or automatically.

[0088] In one or more embodiments, the conflict review includes identifying a conflict in which control loop data for two different pairings are the same and are paired respectively with different templates and reviewing the digitized design data to identify an additional attribute of the respective first and second control loops that is different for the first control loop relative to the second control loop. The review data can include a new feature to be added to the control loop data for the first and second control loops that corresponds to the additional attribute so that the first and second control loops have different corresponding control loop data. The method can further include adding the new feature to the control loop data.

[0089] In the preceding, reference is made to various embodiments. However, the scope of the present disclosure is not limited to the specific described embodiments. Instead, any combination of the described features and elements, whether related to different embodiments or not, is contemplated to implement and practice contemplated embodiments. Furthermore, although embodiments may achieve advantages over other possible solutions or over the prior art, whether or not a particular advantage is achieved by a given embodiment is not limiting of the scope of the present disclosure. Thus, the preceding aspects, features, embodiments and advantages are merely illustrative and are not considered elements or limitations of the appended claims except where explicitly recited in a claim(s).

[0090] The various embodiments disclosed herein may be implemented as a system, method or computer program product. Accordingly, aspects may take the form

of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, aspects may take the form of a computer program product embodied in one or more computer-readable medium(s) having computer-readable program code embodied thereon.

[0091] Any combination of one or more computer-readable medium(s) may be utilized. The computer-readable medium may be a non-transitory computer-readable medium. A non-transitory computer-readable medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the non-transitory computer-readable medium can include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, RAM, a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. Program code embodied on a computer-readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

[0092] Computer program code for carrying out operations for aspects of the present disclosure may be written in any combination of one or more programming languages. Moreover, such computer program code can execute using a single computer system or by multiple computer systems communicating with one another (e.g., using a LAN, WAN, the Internet, etc.). While various features in the preceding are described with reference to flowchart illustrations and/or block diagrams, a person of ordinary skill in the art will understand that each block of the flowchart illustrations and/or block diagrams, as well as combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer logic (e.g., computer program instructions, hardware logic, a combination of the two, etc.). Generally, computer program instructions may be provided to a processor(s) of a general-purpose computer, special-purpose computer, or other programmable data processing apparatus. Moreover, the execution of such computer program instructions using the processor(s) produces a machine that can carry out a function(s) or act(s) specified in the flowchart and/or block diagram block or blocks.

[0093] Unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this disclosure belongs. Although any methods and materials similar or equivalent to those described herein can also be used in the practice or testing of the illustrated

embodiments, exemplary methods and materials are now described. All publications mentioned herein are incorporated herein by reference to disclose and describe the methods and/or materials in connection with which the publications are cited.

[0094] It must be noted that as used herein and in the appended claims, the singular forms "a", "an," and "the" include plural referents unless the context clearly dictates otherwise. Thus, for example, reference to "a stimulus" includes a plurality of such stimuli and reference to "the signal" includes reference to one or more signals and equivalents thereof known to those skilled in the art, and so forth.

[0095] It is to be appreciated the embodiments of the disclosure include software algorithms, programs, or code that can reside on a computer useable medium having control logic for enabling execution on a machine having a computer processor. The machine typically includes memory storage configured to provide output from execution of the computer algorithm or program.

[0096] As used herein, the term "software" is meant to be synonymous with any code or program that can be in a processor of a host computer, regardless of whether the implementation is in hardware, firmware or as a software computer product available on a disc, a memory storage device, or for download from a remote machine. The embodiments described herein include such software to implement the logic, equations, relationships and algorithms described above. One skilled in the art will appreciate further features and advantages of the illustrated embodiments based on the above-described embodiments. Accordingly, the illustrated embodiments are not to be limited by what has been particularly shown and described, except as indicated by the appended claims.

[0097] Embodiments of design system 100 may be implemented or executed by one or more computer systems. For example, design system 100 can be implemented using a computer system such as example computer system 1100 illustrated in FIG. 11. In various embodiments, computer system 1100 may be a server, a mainframe computer system, a workstation, a network computer, a desktop computer, a laptop, or the like, and/or include one or more of a field-programmable gate array (FPGA), application specific integrated circuit (ASIC), microcontroller, microprocessor, or the like.

[0098] Computer system 1100 is only one example of a suitable system and is not intended to suggest any limitation as to the scope of use or functionality of embodiments of the disclosure described herein. Regardless, computer system 1100 is capable of being implemented and/or performing any of the functionality set forth hereinabove.

[0099] Computer system 1100 may be described in the general context of computer system-executable instructions, such as program modules, being executed by a computer system. Generally, program modules may include routines, programs, objects, components, logic, data structures, and so on that perform particular tasks

or implement particular abstract data types. Computer system 1100 may be practiced in distributed data processing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed data processing environment, program modules may be located in both local and remote computer system storage media including memory storage devices.

[0100] Computer system 1100 is shown in FIG. 11 in the form of a general-purpose computing device. The components of computer system 1100 may include, but are not limited to, one or more processors or processing units 1116, a system memory 1128, and a bus 1118 that couples various system components including system memory 1128 to processor 1116.

[0101] Bus 1118 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus.

[0102] Computer system 1100 typically includes a variety of computer system readable media. Such media may be any available media that is accessible by computer system 1100, and it includes both volatile and non-volatile media, removable and non-removable media.

[0103] System memory 1128 can include computer system readable media in the form of volatile memory, such as random-access memory (RAM) 1130 and/or cache memory 1132. Computer system 1100 may further include other removable/non-removable, volatile/non-volatile computer system storage media. By way of example only, storage system 1134 can be provided for reading from and writing to a non-removable, non-volatile magnetic media (not shown and typically called a "hard drive"). Although not shown, a magnetic disk drive for reading from and writing to a removable, non-volatile magnetic disk, and an optical disk drive for reading from or writing to a removable, non-volatile optical disk such as a CD-ROM, DVD-ROM or other optical media can be provided. In such instances, each can be connected to bus 1118 by one or more data media interfaces. As will be further depicted and described below, memory 1128 may include at least one program product having a set (e.g., at least one) of program modules that are configured to carry out the functions of embodiments of the disclosure.

[0104] Program/utility 1140, having a set (at least one) of program modules 1115, such as computer system model 120, filter 122, spatial positioning module 124, analysis module 126, and user interface 128 may be stored in memory 1128 by way of example, and not limitation, as well as an operating system, one or more application programs, other program modules, and pro-

gram data. Each of the operating system, one or more application programs, other program modules, and program data or some combination thereof, may include an implementation of a networking environment. Program modules 1115 generally carry out the functions and/or methodologies of embodiments of the disclosure as described herein.

[0105] Computer system 1100 may also communicate with one or more external devices 1114 such as a keyboard, a pointing device, a display 1124, etc.; one or more devices that enable a user to interact with computer system 1100; and/or any devices (e.g., network card, modem, etc.) that enable design system 100 to communicate with one or more other computing devices. Such communication can occur via Input/Output (I/O) interfaces 1122. Still yet, computer system 1100 can communicate with one or more networks such as a LAN, a general WAN, and/or a public network (e.g., the Internet) via network adapter 1120. As depicted, network adapter 1120 communicates with the other components of a network (not shown) via bus 1118. It should be understood that although not shown, other hardware and/or software components could be used in conjunction with computer system 1100. Examples, include, but are not limited to: microcode, device drivers, redundant processing units, external disk drive arrays, RAID systems, tape drives, and data archival storage systems, and so on.

[0106] The flowchart and block diagrams in the Figures illustrate the architecture, functionality and/or operation of possible implementations of various embodiments of the present disclosure. In this regard, each block in the flowchart or block diagrams may represent a module, segment or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0107] It is to be understood that the above description is intended to be illustrative, and not restrictive. Many other implementation examples are apparent upon reading and understanding the above description. Although the disclosure describes specific examples, it is recognized that the systems and methods of the disclosure are not limited to the examples described herein, but may be practiced with modifications within the scope of the appended claims. Accordingly, the specification and drawings are to be regarded in an illustrative sense rather than a restrictive sense. The scope of the disclosure

should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

Claims

1. A method of building a knowledge base, the method comprising:

receiving a training set including pairings of control loop data for respective control loops identified in digitized design data and templates that were instantiated using the control loop data of the respective control loops; and training, using machine learning, a knowledge base, based on the training set, wherein the knowledge base, once trained, is configured to be queried with digitized new control loop data, predict a template to pair with the digitized new control loop data, and the predicted template, and the predicted template is configured to be instantiated with the new control loop data for implementation of a control loop in an engineering system.

2. The method of claim 1, wherein the knowledge base includes a machine learning model, and training the machine learning model includes training a classifier of the machine learning model.

3. The method of claim 1, wherein the received digitized control loop data and the received templates are standardized and/or normalized.

4. The method of claim 1, further comprising:

submitting the pairings for a conflict review performed manually and/or automatically; receiving review data based on the conflict review; and updating the control loop data in the pairings as a function of the conflict review data, wherein the knowledge base is trained using the updated pairings.

5. The method of claim 4, wherein the conflict review comprises:

identifying a conflict in which first and second control loop data for two different pairings are the same and are paired respectively with different templates; reviewing the digitized design data to identify an additional attribute of the respective first and second control loops that is different for the first control loop relative to the second control loop; and

- including in the review data a new feature to be added to the control loop data for the first and second control loops that corresponds to the additional attribute so that the first and second control loops have different corresponding control loop data.
6. The method of claim 5, further comprising adding the additional attribute to the control loop data of at least one of the first and second control loops.
7. The method of claim 1, further comprising:
- receiving feedback about pairings between digitized new control loop data and predicted templates output by the knowledge base in response to queries submitted to the knowledge base; and
- updating the training set based on the feedback.
8. The method of claim 7, wherein the knowledge based further configured to adjust a confidence score associated with the predicted template based on the feedback.
9. A machine learning system for building a knowledge base for use with an automated engineering system, the machine system comprising:
- a memory configured to store instructions; and
- a processor and in communication with the memory, wherein the processor upon execution of the instructions is configured to: receive a training set including pairings of control loop data for respective control loops identified in digitized design data and templates that were instantiated using the control loop data of the respective control loops; and
- train, using machine learning, a knowledge base, based on the training set, wherein the knowledge base, once trained, is configured to be queried with digitized new control loop data, predict a template to pair with the digitized new control loop data, and the predicted template, and the predicted template is configured to be instantiated with the new control loop data for implementation of a control loop in an engineering system.
10. The machine learning system of claim 9, wherein the knowledge base includes a machine learning model, and training the machine learning model includes training a classifier of the machine learning model.
11. The machine learning system of claim 9, wherein the received digitized control loop data and the received
- templates are standardized and/or normalized.
12. The machine learning system of claim 9, wherein the processor upon execution of the instructions is further configured to:
- submitting the pairings for a conflict review performed manually and/or automatically;
- receiving review data based on the conflict review; and
- updating the control loop data in the pairings as a function of the conflict review data, wherein the knowledge base is trained using the updated pairings.
13. The machine learning system of claim 12, wherein the conflict review comprises:
- identifying a conflict in which first and second control loop data for two different pairings are the same and are paired respectively with different templates;
- reviewing the digitized design data to identify an additional attribute of the respective first and second control loops that is different for the first control loop relative to the second control loop; and
- including in the review data a new feature to be added to the control loop data for the first and second control loops that corresponds to the additional attribute so that the first and second control loops have different corresponding control loop data.
14. The machine learning system of claim 13, wherein the processor upon execution of the instructions is further configured to add the additional attribute to the control loop data of at least one of the first and second control loops.
15. The machine learning system of claim 9, wherein the processor upon execution of the instructions is further configured to:
- receive feedback FROM ENGINEERING REVIEW about pairings between digitized new control loop data and predicted templates output by the knowledge base in response to queries submitted to the knowledge base; and
- updating the training set based on the feedback.
16. The machine learning system of claim 15, wherein the knowledge based further configured to adjust a confidence score associated with the predicted template based on the feedback.
17. A non-transitory computer readable storage medium having one or more computer programs stored there-

in, the computer programs comprising instructions, which when executed by a processor of a computer system, cause the processor to: receiving a training set including pairings of control loop data for respective control loops identified in digitized design data and templates that were instantiated using the control loop data of the respective control loops; and

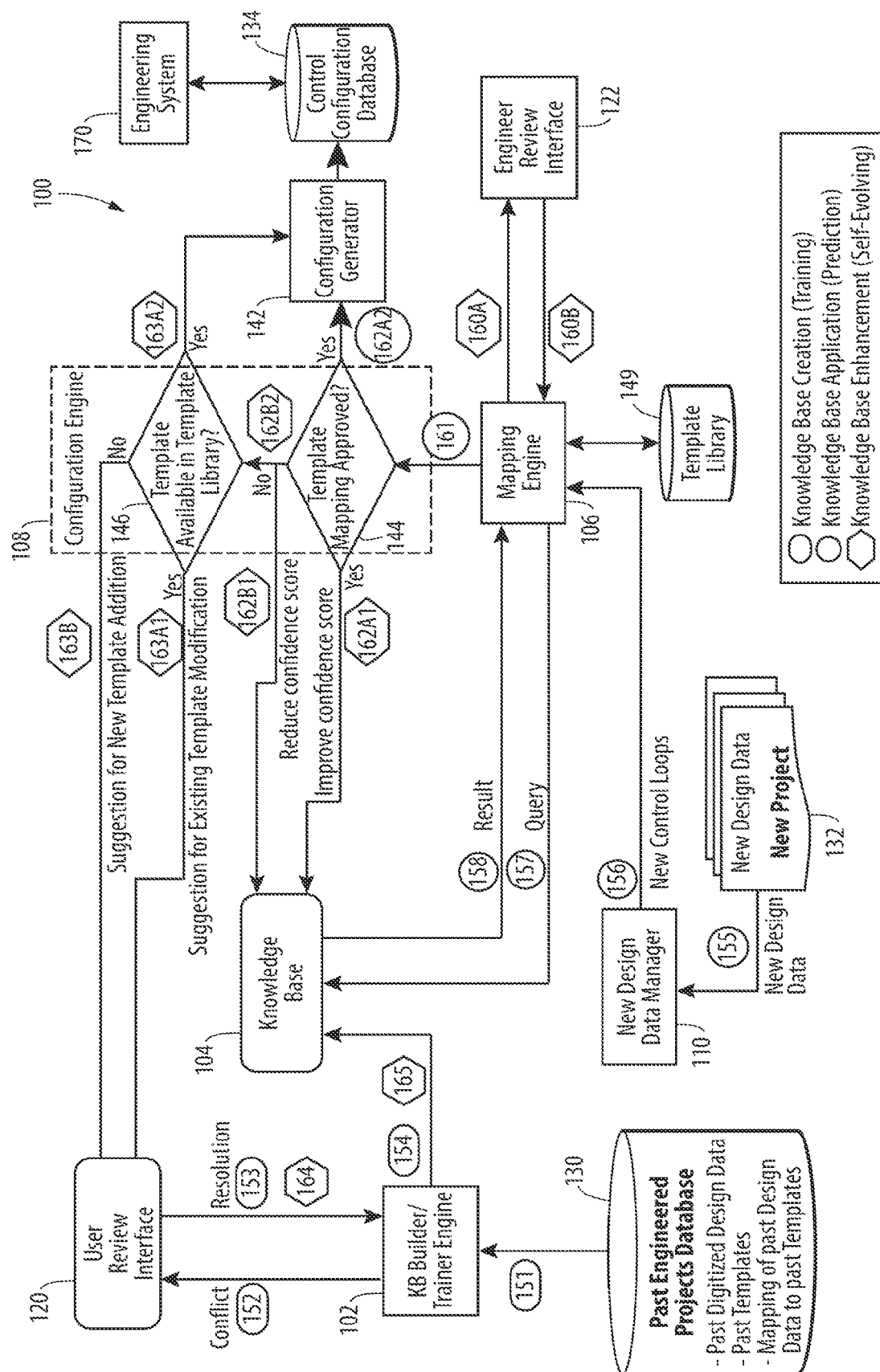
training, using machine learning, a knowledge base, based on the training set, wherein the knowledge base, once trained, is configured to be queried with digitized new control loop data, predict a template to pair with the digitized new control loop data, and the predicted template, and the predicted template is configured to be instantiated with the new control loop data for implementation of a control loop in an engineering system.

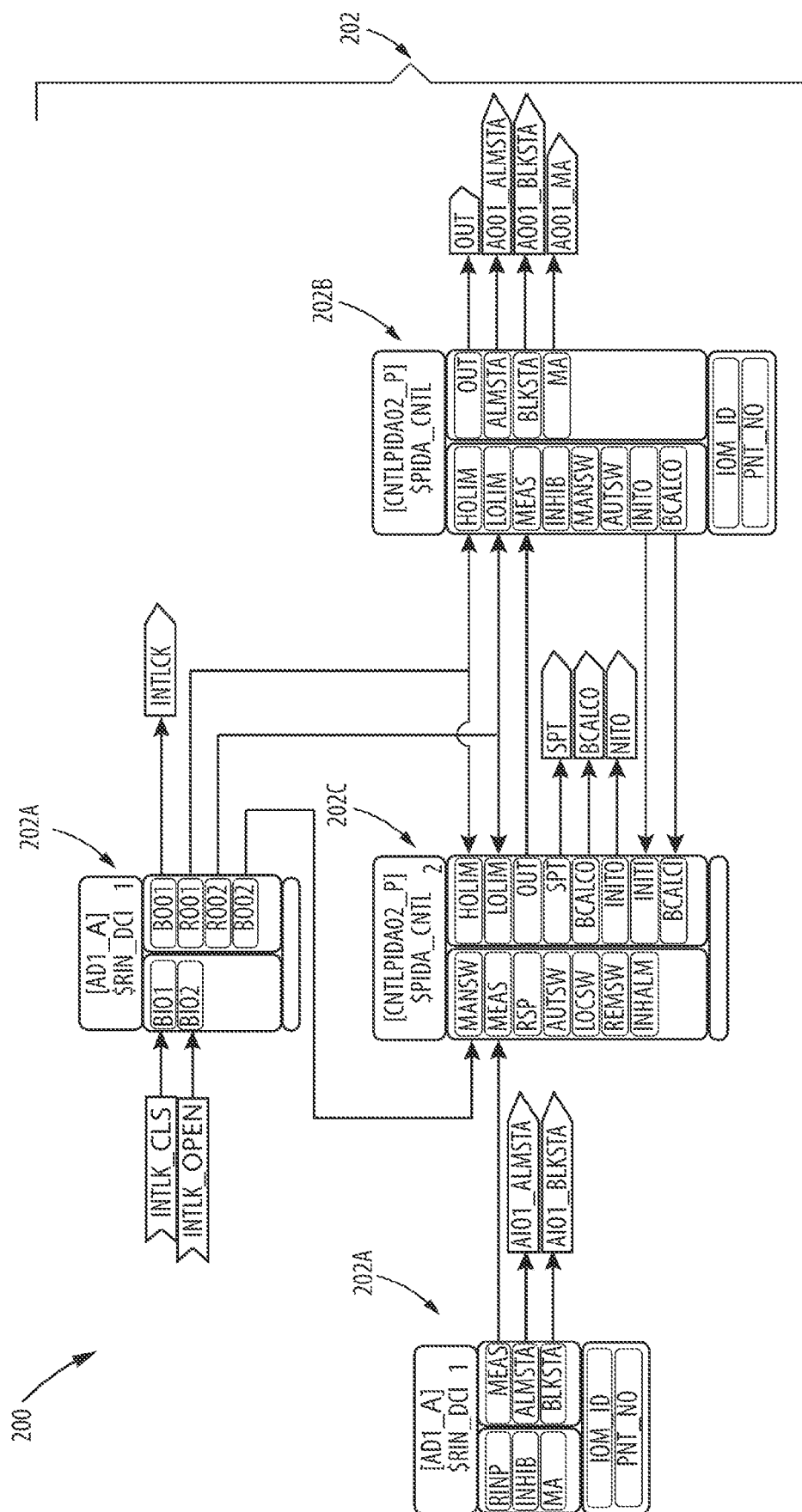
18. The non-transitory computer readable storage medium of claim 17, wherein the knowledge base includes a machine learning model, and training the machine learning model includes training a classifier of the machine learning model.

19. The non-transitory computer readable storage medium of claim 17, wherein the computer programs comprising instructions, when executed by a processor of a computer system, further cause the processor to:

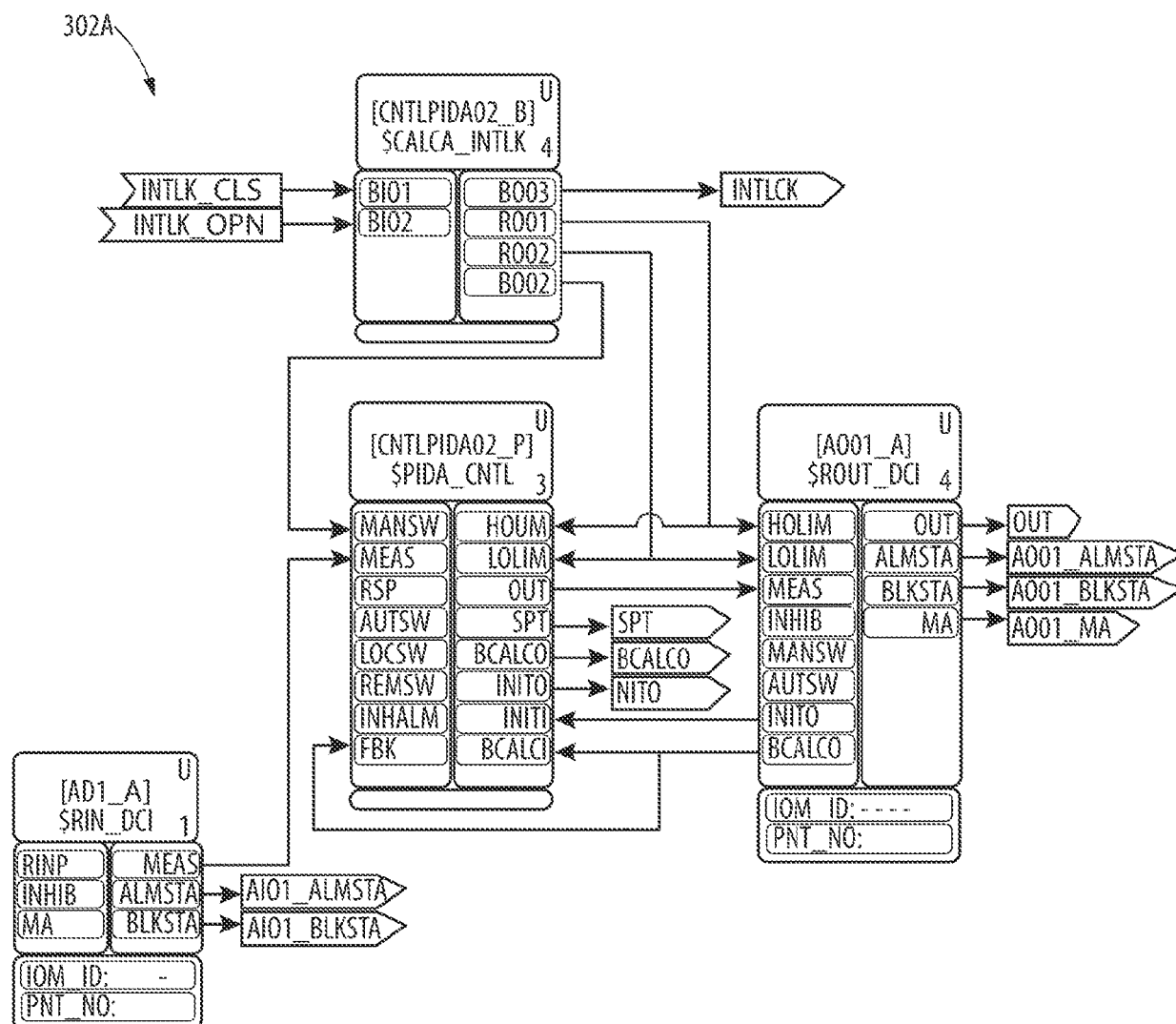
receive feedback about pairings between digitized new control loop data and predicted templates output by the knowledge base in response to queries submitted to the knowledge base; and
update the training set based on the feedback.

20. The non-transitory computer readable storage medium of claim 19, wherein the knowledge based further configured to adjust a confidence score associated with the predicted template based on the feedback.



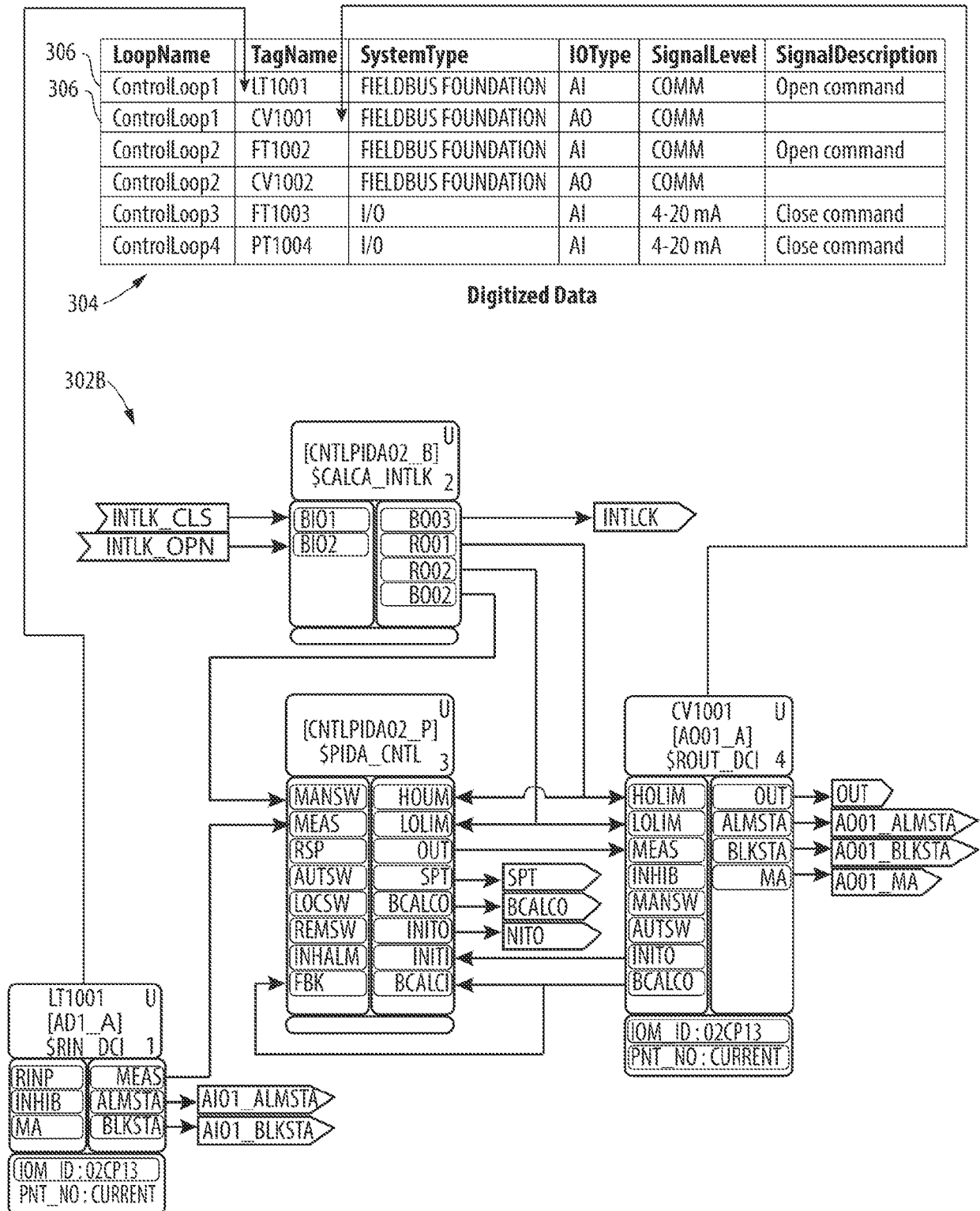


2
9
11



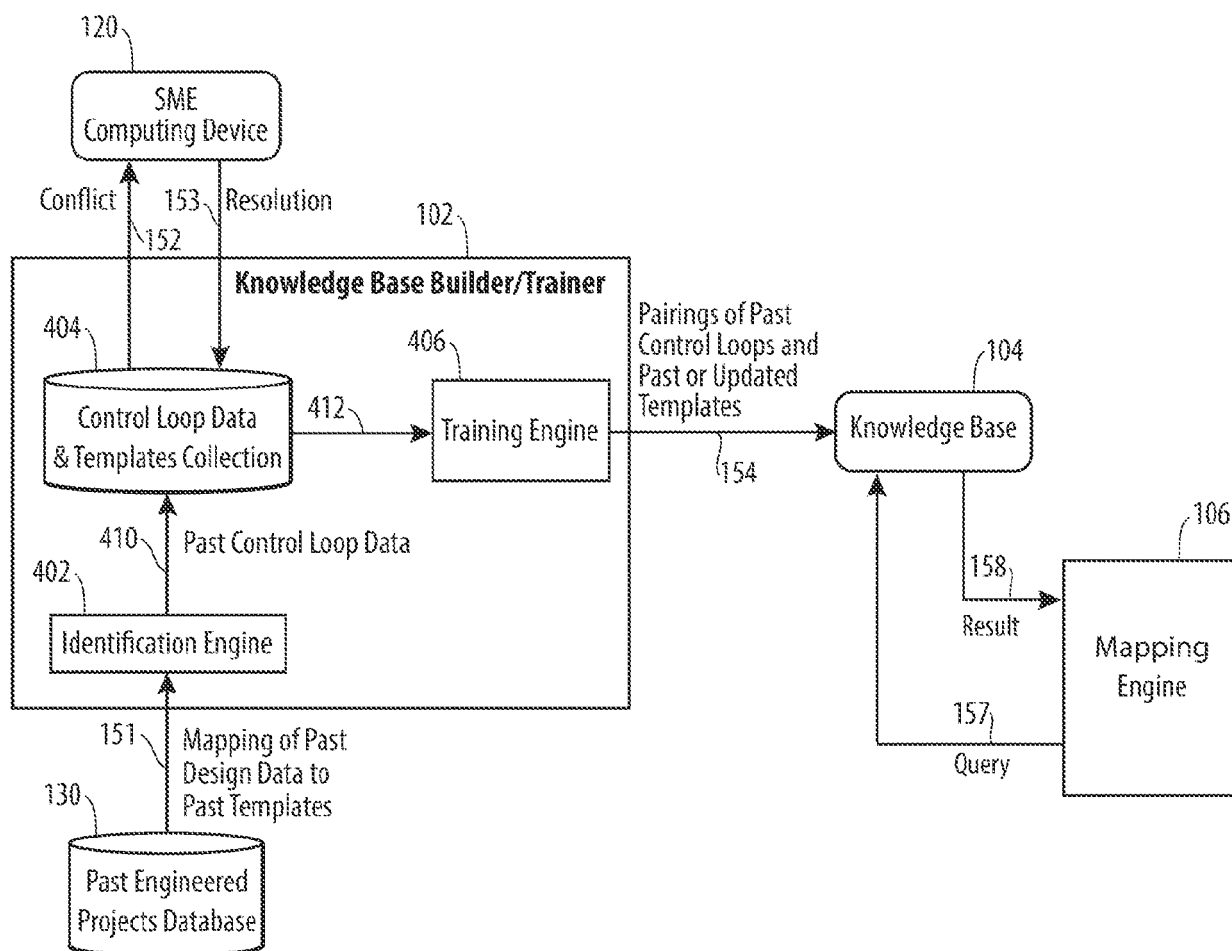
Template before mapping

Fig. 3



Control configuration - Template mapped (instantiated)

Fig. 3
(Cont'd)



Knowledge Base Creation and Application

Fig. 4

404

504

502

Rules auto created based on the characteristics identified for
the control loops by scanning data across various sources TagDB, P&ID, ...

510

500

506

508

Control Loop	SystemType	IoType	SignalLevel	SignalDescription/Add a Feature	Target Template	Conflict Free Confirmation			
ControlLoop1, ControlLoop2	FIELDBUS FOU	AI - 1 AO - 1	COMM-2	Open command - 1	SF_AI_FF_1AI	Accept/Reject ☐ OK ☐			
ControlLoop3, ControlLoop4	I/O	AI	4-20 mA (Ext. power supply)	Close command	SF_AI_HW_1AI	☐ OK ☐			
ControlLoop5, ControlLoop6	I/O SPY	DI	Dry Contact	Stop command	\$CIN	☐ OK ☐			
ControlLoop7	MODBUS TCP	COMM. COMM. COMM., COMM. COMM. COMM.							
		AI, AI, AI, AI, AI, AI, AI, AI,							
		AI, AI, AI, AI, DI, AI, AO, DI, DI, DI, DI, DI, DI							
			4-20 ,A (Ext. power supply), 4-20 mA Loop powered		SDuty_789	☐ OK ☐			

404

502

504

510

600A

506

508

Identified characteristics of control loops by scanning the Customer design data (TagDB, P&ID, ...)							Conflict Free Confirmation
Control Loop	SystemType	IoType	SignalLevel	SignalDescription	Add a Feature	StrategyTemplateName	Accept/Reject
ControlLoop8	FIELD BUS FOUNDATION	AI	COMM	Open command		SF_AI_FF_1AI	☐ NO ☐
ControlLoop9	FIELD BUS FOUNDATION	AI	COMM	Open command		SF_AI_FF_FT_1AI	☐ NO ☐
ControlLoop10	I/O SFY	DI	Dry Contact	Stop command		SCIN	☐ OK ☐

502

502										504										510										506										508									

Fig. 6

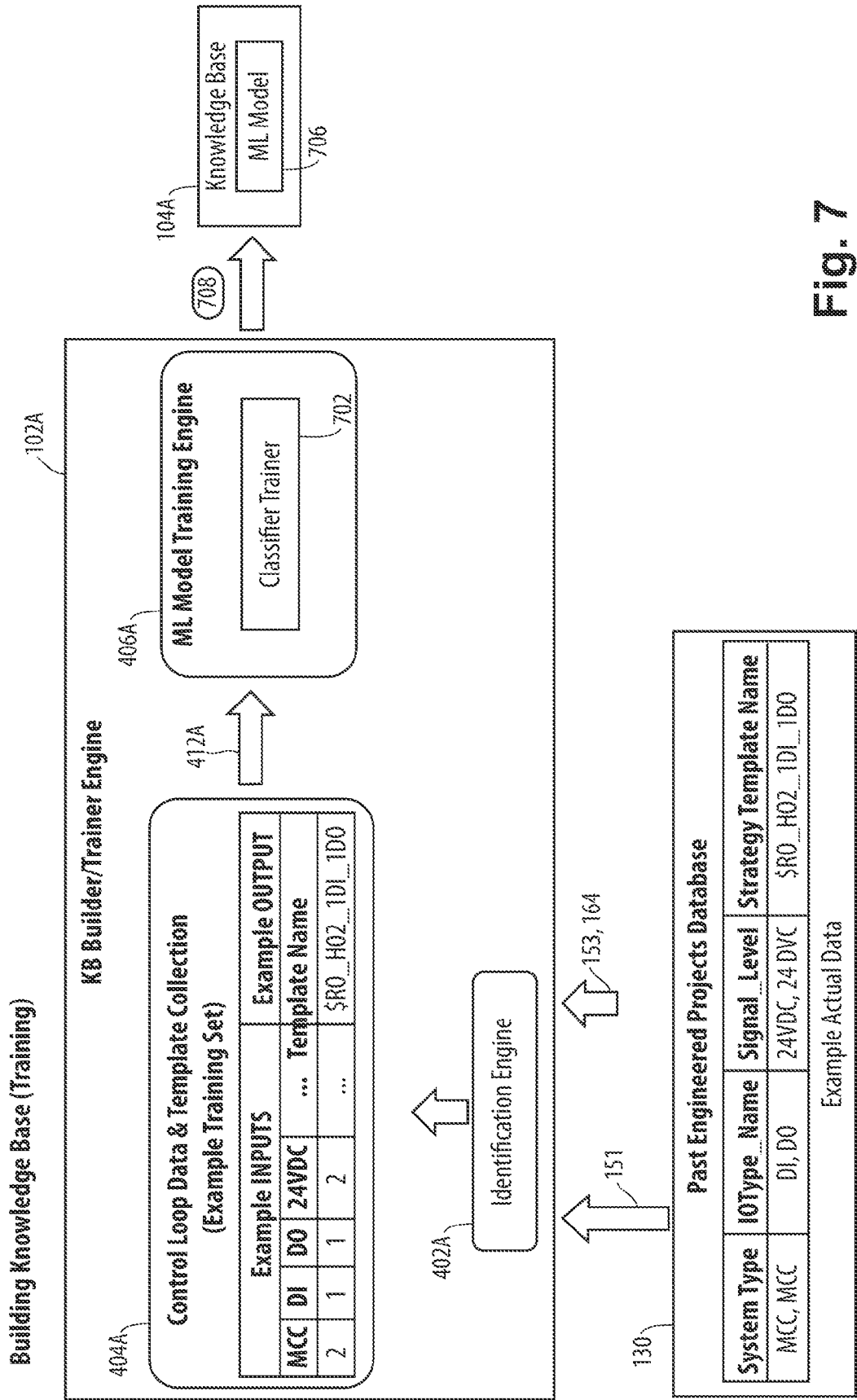


Fig. 7

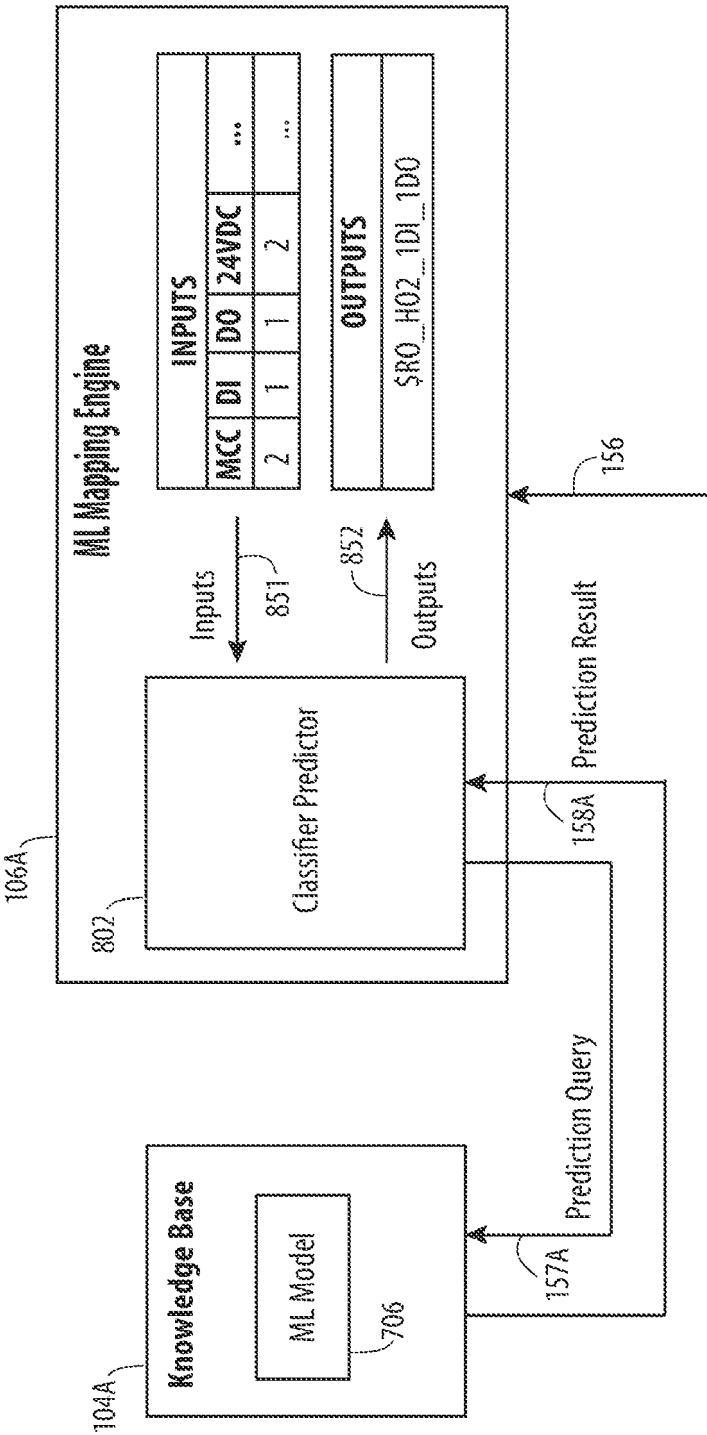
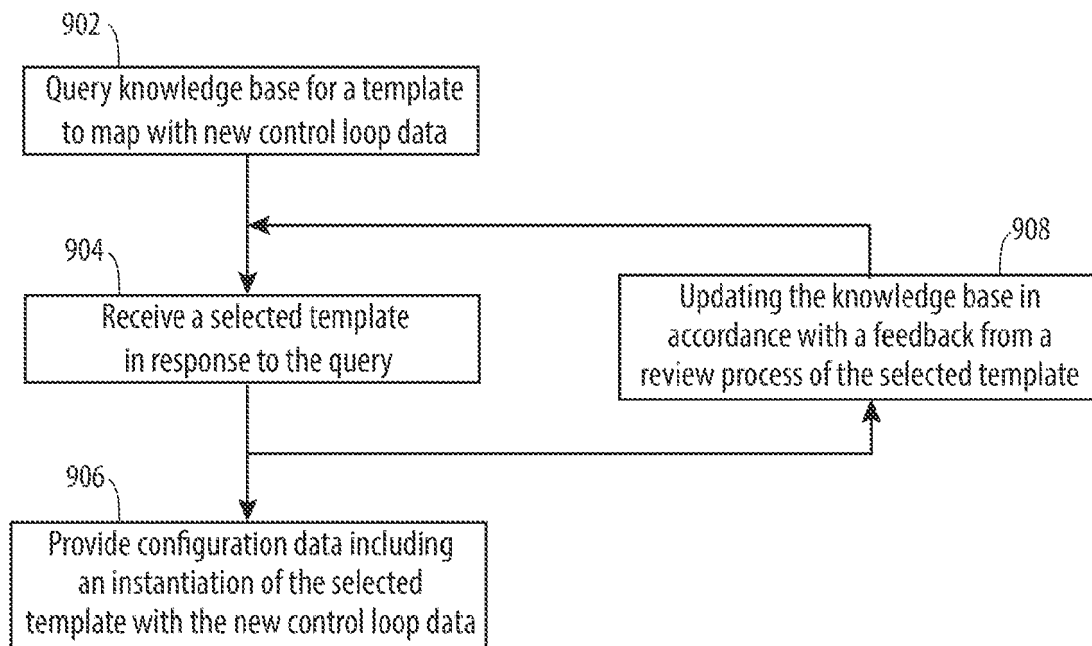
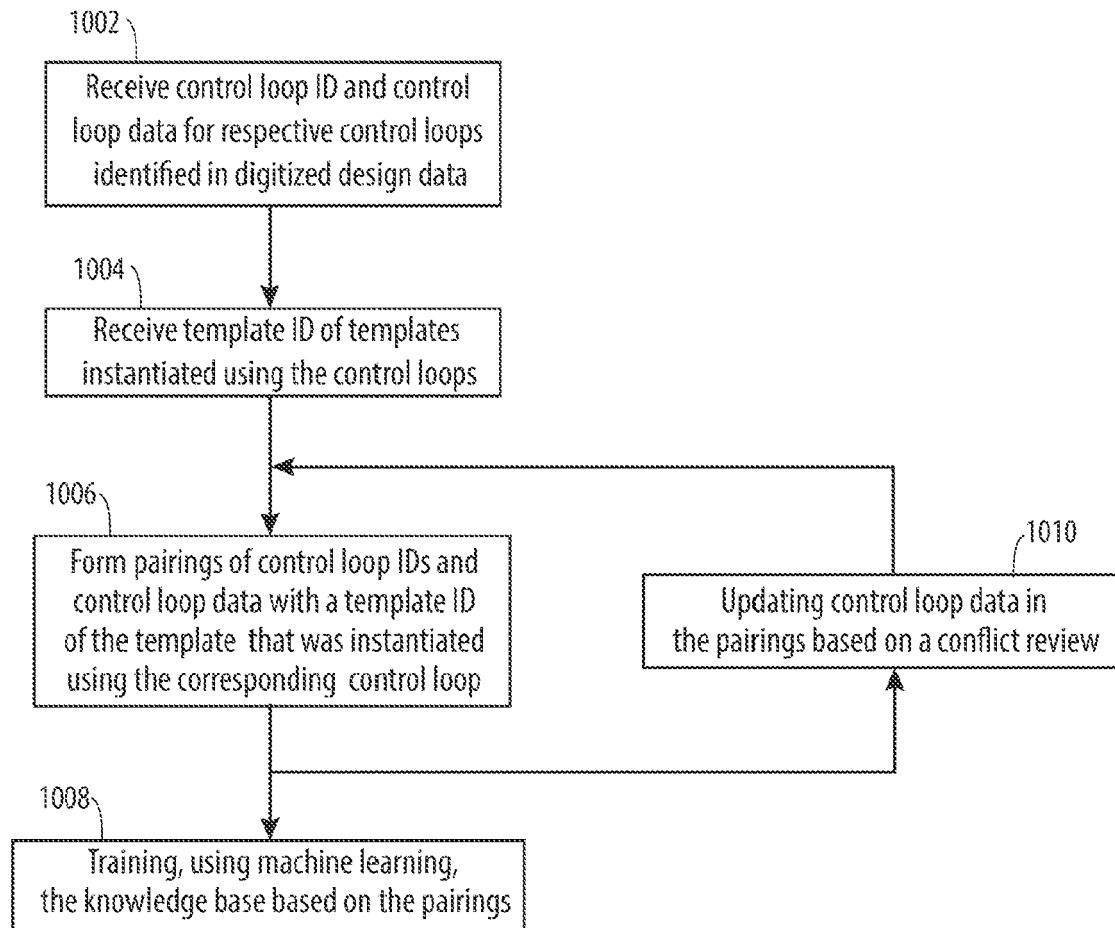


Fig. 8

**Fig. 9**

**Fig. 10**

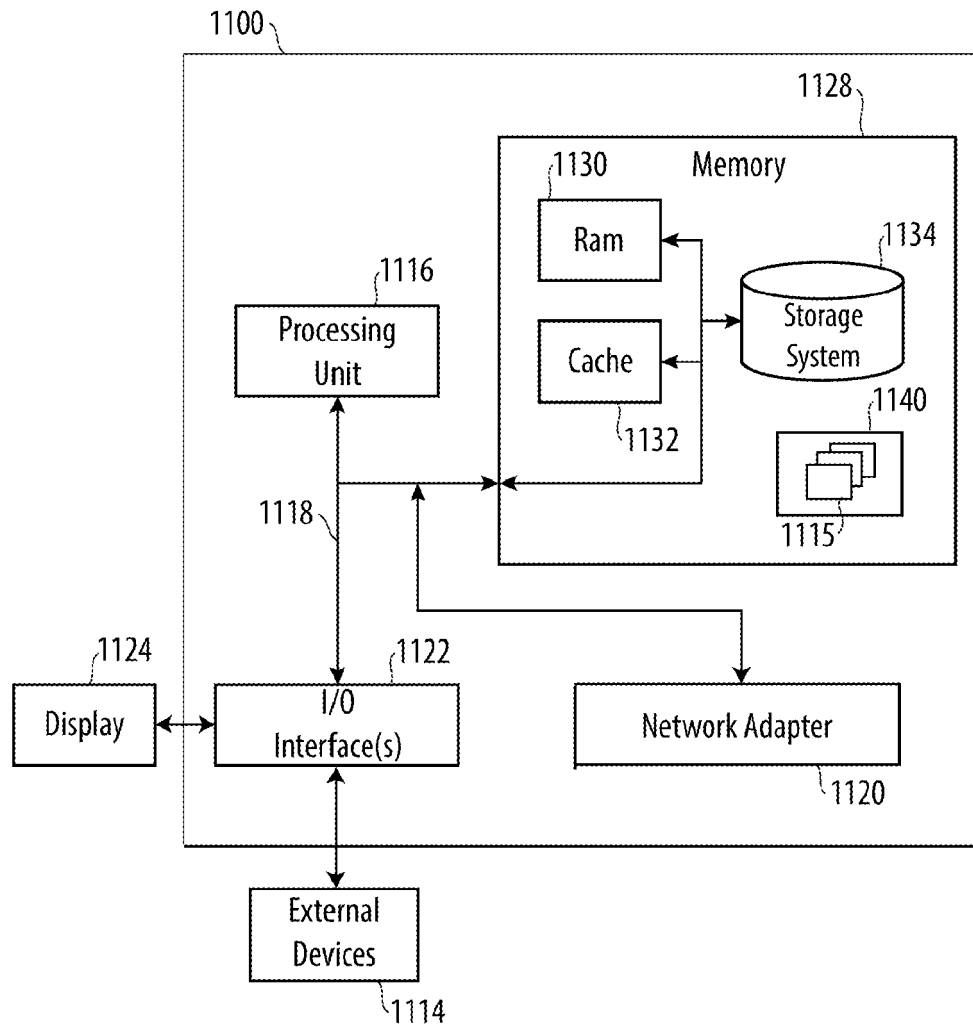


Fig. 11



EUROPEAN SEARCH REPORT

Application Number

EP 22 21 0980

5

10

15

20

25

30

35

40

45

50

55

1

EPO FORM 1503 03.82 (P04C01)

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (IPC)
X	WO 2019/164503 A1 (SIEMENS AG [DE]) 29 August 2019 (2019-08-29) * claims 1-8 * * figures 1-4 * * paragraph [0001] - paragraph [0019] * -----	1-20	INV. G06N5/022 G06N20/00 G06N5/04 G05B19/042 G06N3/0442
A	US 2020/159836 A1 (ALKAN OZNUR [IE] ET AL) 21 May 2020 (2020-05-21) * claims 1-7 * * paragraph [0070] - paragraph [0096] * -----	1-20	G06N5/01 G06N20/20 G06N3/09
A	WO 2020/198249 A1 (SCHNEIDER ELECTRIC SYSTEMS USA INC [US]) 1 October 2020 (2020-10-01) * the whole document * -----	1-20	
			TECHNICAL FIELDS SEARCHED (IPC)
			G06N G05B G06F
The present search report has been drawn up for all claims			
Place of search Munich		Date of completion of the search 15 May 2023	Examiner Rackl, Günther
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO.

EP 22 21 0980

5

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

15-05-2023

10

15

20

25

30

35

40

45

50

55

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2019164503 A1	29-08-2019	NONE	
US 2020159836 A1	21-05-2020	NONE	
WO 2020198249 A1	01-10-2020	AU 2020247915 A1	14-10-2021
		AU 2020247916 A1	14-10-2021
		AU 2020248388 A1	14-10-2021
		CA 3131421 A1	01-10-2020
		CA 3133312 A1	01-10-2020
		CA 3134938 A1	01-10-2020
		CN 113614655 A	05-11-2021
		CN 113614656 A	05-11-2021
		CN 113614666 A	05-11-2021
		CN 113853553 A	28-12-2021
		EP 3924790 A1	22-12-2021
		EP 3928168 A1	29-12-2021
		EP 3935453 A1	12-01-2022
		EP 3935460 A1	12-01-2022
		JP 2022521816 A	12-04-2022
		JP 2022521817 A	12-04-2022
		JP 2022524642 A	09-05-2022
		US 2022171891 A1	02-06-2022
		US 2022187815 A1	16-06-2022
		US 2022187818 A1	16-06-2022
		US 2022197272 A1	23-06-2022
		WO 2020198249 A1	01-10-2020
		WO 2020198250 A1	01-10-2020
		WO 2020198254 A1	01-10-2020
		WO 2020198256 A1	01-10-2020

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 10671894 B [0034] [0053]
- US 10635945 B [0034] [0053]
- US 442403 [0036]
- US 17442423 [0036]
- US 17442437 [0036]